TASK FORCE WEB

Web Services Architecture
9 July 2002

D. Taylor Holmes, P.D.I.T.
Senior Engineer
SPAWAR Systems Center Charleston

Agenda

- Web Services Architecture
  - Overview
  - Architecture Components (SOAP, WSDL, UDDI)
- SOAP 1.1 Overview
  - Features
  - SOAP Messages (Diagram, Example)
- WSDL 1.0 Overview
  - Features
  - WSDL Documents (Example)
- UDDI 2.0 Overview
  - Features
  - SOAP\WSDL\UDDI Integration Demo

Web Services Architecture

- Overview
  - Definition: Web Services are modular applications that are self-describing and that can be published, located and invoked over the web based on open Internet standards
  - Standards based with XML at the core
  - Overcomes limitations of proprietary platforms to connect disparate applications at the business logic and data level
  - Foundation framework for the semantic web

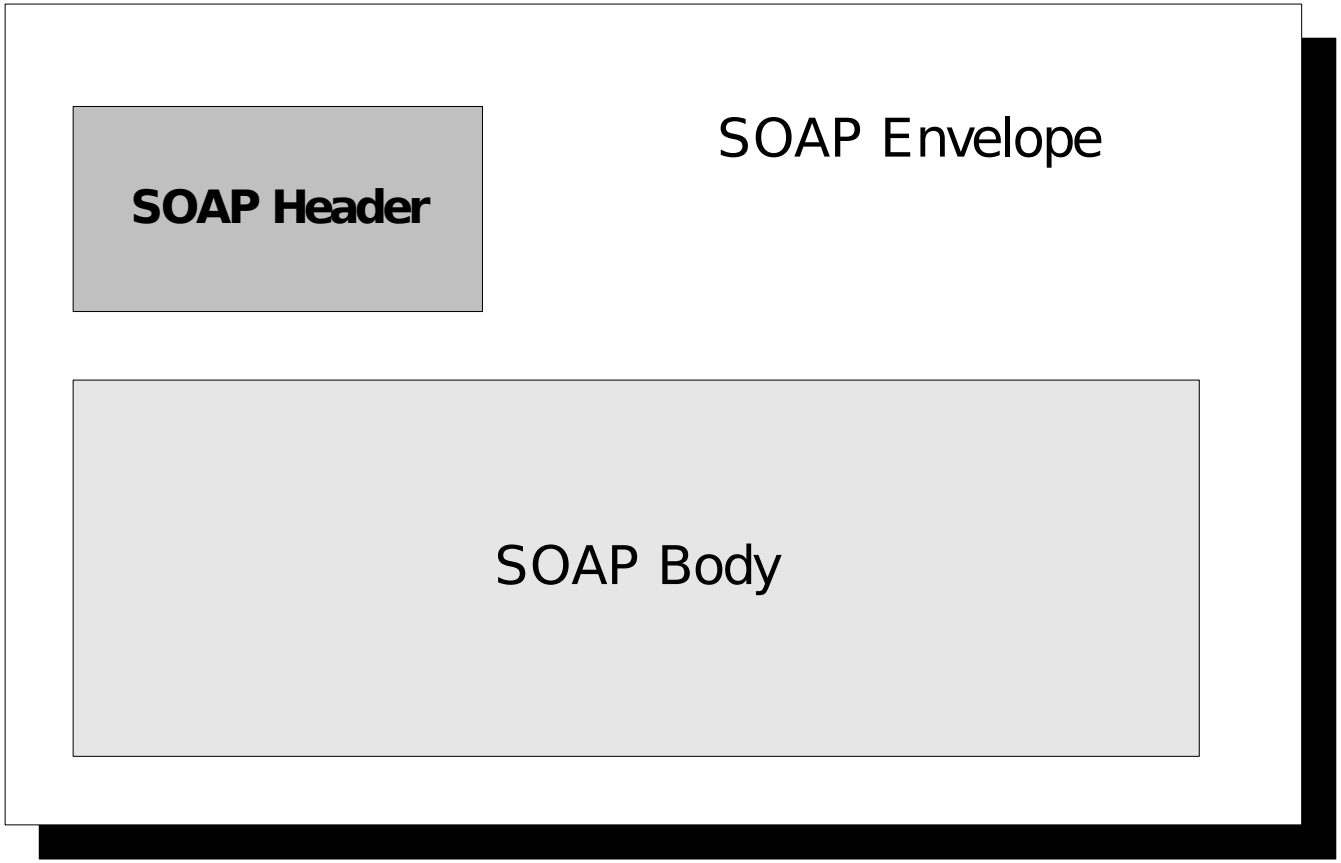
Architecture Components

- SOAP (Simple Object Access Protocol)
  - An XML based communications protocol for exchanging information between heterogeneous computing platforms
  - Designed to solve the functional problem of Invocation
- WSDL (Web Service Description Language)
  - An XML based standard for describing the interface of a Web Service i.e. the messages that a Web Service can accept and generate.
  - Designed to solve the functional problem of Description
- UDDI (Universal Description, Discovery and Integration)
  - An XML based specification that describes how a provider can advertise the existence of their Web Services in a registry.
  - Designed to solve the functional problem of Discovery

SOAP 1.1 Overview

- Features
  - Simple to implement using at a minimum an XML Parser, a few lines of code and an HTTP Server
  - Extensible via the use of XML
  - Uses existing Internet infrastructure (common transport protocols such as HTTP, firewall friendly)
- SOAP Messages
  - SOAP Messages are fundamentally one-way transmissions from a sender to a receiver
  - SOAP Messages are XML Documents with their own schema

SOAP Message Diagram



SOAP Message Example

- Simple SOAP Message that invokes the get_businessDetail UDDI API operation of the Service Registry with one parameter - businessKey

```
<SOAP-ENV:Envelope xmlns:SOAP-
    ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
    <m:get_businessDetail xmlns:m="urn:ServiceRegistry">
        <businessKey>8AE56BB5-0916-49E0-A359-
        9E69A7D0C8CB</businessKey>/>
    </m:get_businessDetail>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

WSDL 1.0 Overview

- Features
  - Support the ability for Web Services to be Self-Describing
  - Functionally equivalent to Reflection in Java and Type Libraries in COM
  - Will likely be generated by a tool
- WSDL Documents
  - Composed of definitions, types, message, operation, portType, binding, port and service elements

WSDL Document Example

```
<definitions targetNamespace="http://sc-taylor/serviceregistry/webservices/serviceregistry.wsdl"
    xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-
    instance" xmlns:tns="http://sc-taylor /serviceregistry/webservices/serviceregistry.wsdl"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      <types>
          <schema targetNamespace="http://sc-
          taylor/serviceregistry/webservices/serviceregistry.wsdl"
          xmlns="http://www.w3.org/2000/10/XMLSchema"/>
      </types>
      <message name="get_businessDetailRequest">
          <part name="businessKey" type="xsd:string"/>
      </message>
      <message name="get_businessDetailResponse">
          <part name="return" type="xsd:string"/>
      </message>
      <portType name="ServiceRegistryPortType">
            <operation name="get_businessDetail">
                  <input message="tns:get_businessDetailRequest"/>
                  <output message="tns:get_businessDetailResponse"/>
            </operation>
      </portType>
```

WSDL Document Example contd.

```
<binding name="ServiceRegistryBinding" type="tns:ServiceRegistryPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation name="get_businessDetail">
        <soap:operation soapAction="urn:get_businessDetail"/>
        <input>
            <soap:body use="encoded" namespace="urn:ServiceRegistry"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
        </input>
        <output>
            <soap:body use="encoded" namespace="urn:ServiceRegistry"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
        </output>
    </operation>
</binding>
<service name="ServiceRegistry">
    <documentation>Implements the UDDI SOAP API for the Service
    Registry</documentation>
    <port name="ServiceRegistryPort" binding="tns:ServiceRegistryBinding">
    <soap:address location="http://sc-taylor/ServiceRegistry/webservices/uddiAPI.asp"/>
    </port>
</service>
</definitions>
```

UDDI Overview

- Features
  - Publish – specifies how a Web Service provider to register itself and it’s services
  - Find – specifies how a client application finds the description of a Web Service or provider
  - Bind – specifies how a client application connects to and interacts with the Web Service after finding it

UDDI Overview

- SOAP\WSDL\UDDI Integration Demo
  - Use WSEE to dynamically read the Service Registry WSDL document
  - Generate a SOAP Client Message to invoke the get_businessDetails operation
  - Invoke the operation and receive a SOAP Response Message with XML\XSL payload.
  - Show format of the SOAP Response Message and the resulting XML\XSL payload.